

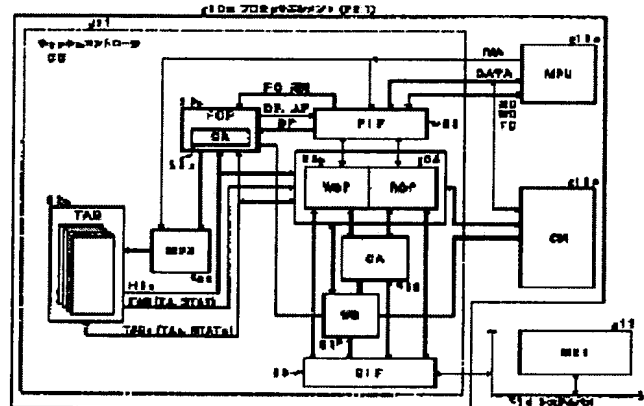
## ELECTRONIC COMPUTER WITH CACHE MEMORY

<b>Patent number:</b>	JP8095861
<b>Publication date:</b>	1996-04-12
<b>Inventor:</b>	AZEZAKI TSUTOMU
<b>Applicant:</b>	TOKYO SHIBAURA ELECTRIC CO
<b>Classification:</b>	
- international:	G06F12/08
- european:	
<b>Application number:</b>	JP19940227617 19940922
<b>Priority number(s):</b>	JP19940227617 19940922

**Report a data error here**

## Abstract of JP8095861

**PURPOSE:** To execute the processing request from an MPU which is generated during cache flashing operation. **CONSTITUTION:** A cache memory 18a is managed by using a valid bit V and a dirty bit D as a status for managing the cache memory 18a and this electronic computer is equipped with a flash operation block 58 which performs the cache flashing operation for writing the updated data of a cache block back to a memory element 12 by inspecting the cache status and performing a flash execution report DF showing that the cache flashing operation is in processing and a flashing interruption report AF showing that the cache flashing operation is interrupted at an access request from an MPU 16a, and when the MPU 16a generated the access request during the flashing operation, the cache flashing operation is interrupted and access processing by the MPU 16 is performed.



Data supplied from the **esp@cenet** database - Worldwide

**THIS PAGE BLANK (USPTO)**

(11)特許出願公開番号

特開平8-95861

(43)公開日 平成8年(1996)4月12日

(51) Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

### 技術表示箇所

G 0 6 F 12/08

3 1 0 B 7623-5B

審査請求 未請求 請求項の数 8 O L (全 11 頁)

(21)出願番号 特願平6-227617

(22)出願日 平成6年(1994)9月22日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 畦崎 勉

東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

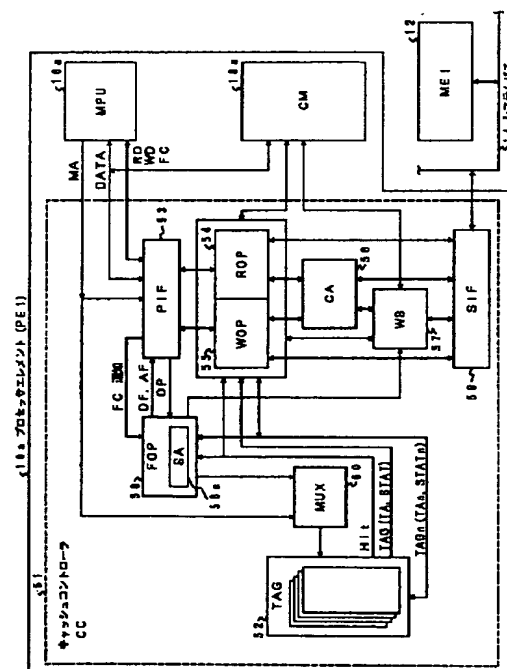
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 キャッシュメモリを有する電子計算機

(57) 【要約】

【目的】キャッシュフラッシュ動作中に発生したMPUからの処理要求を実行することを可能にする。

【構成】 キャッシュメモリ 18 a を管理するステータスとして、有効ビット V、ダーティビット D を用いて管理を行なうものであって、キャッシュステータスを検査して、データが更新されたキャッシュブロックのデータをメモリエlement 12 に書き戻すキャッシュフラッシュを実行すると共に、キャッシュフラッシュ中であることを表すフラッシュ実行通知 DF、及び MPU 16 a からのアクセス要求に応じてキャッシュフラッシュを中断していることを表すフラッシュ中断通知 AF を行なうフラッシュオペレーションブロック 58 を備え、フラッシュ実行中に MPU 16 a によりアクセス要求があった場合に、キャッシュフラッシュを中断して MPU 16 a によるアクセス処理を実行可能にする。



1

## 【特許請求の範囲】

【請求項 1】 演算処理を行なう処理手段と、前記処理手段によって用いられる複数のキャッシュブロックに分割されたキャッシュメモリと、前記キャッシュメモリを制御するキャッシュ制御手段とを備えたプロセッサエレメントが、前記処理手段により扱われる命令及びデータを格納するメモリエレメントとシステムバスを介して接続された電子計算機において、

前記キャッシュ制御手段は、前記キャッシュメモリを管理するステータスとして、キャッシュブロックに有効なデータがあることを示す有効ビット V、キャッシュブロックに格納されているデータが更新されていることを表すダーティビット D を用いて管理を行なうものであって、

前記処理手段からの特殊命令に応じて、全キャッシュブロックに関する前記キャッシュステータスを検査して、データが更新されたキャッシュブロックのデータを前記メモリエレメントに書き戻すキャッシュフラッシュを実行すると共に、キャッシュフラッシュ中であることを表すフラッシュ実行通知、及び前記処理手段からのアクセス要求によってキャッシュフラッシュを中断していることを表すフラッシュ中断通知を行なうフラッシュ手段を備え、

前記フラッシュ手段は、前記フラッシュ実行通知がフラッシュ実行中を示す際に前記処理手段によりアクセス要求があった場合に、キャッシュフラッシュを中断してフラッシュ中断通知を行ない、前記処理手段によるアクセス処理を実行可能にすることを特徴とする電子計算機。

【請求項 2】 前記キャッシュ制御手段は、新たにキャッシュブロックの更新されたデータをメモリエレメントに書き戻したことを表すフラッシュ完了ビット F を、さらに前記ステータスとして設けて前記キャッシュメモリを管理するもので、

前記処理手段からのライトアクセス要求があった場合に、前記ライトアクセス要求の対象となるキャッシュブロックに対してライト処理を実行するライト処理手段と、

前記ライト処理手段によるライト処理の対象となるキャッシュブロックが、ダーティブロックであり、前記フラッシュ手段によるフラッシュの中断前にキャッシュフラッシュが完了したブロックではない場合に、当該キャッシュブロックのデータを前記メモリエレメントに書き戻し、当該キャッシュブロックに対応するステータスのフラッシュ完了ビットをフラッシュ済みにセットする書き戻し手段と、

をさらに具備し、

前記処理手段は、前記書き戻し手段によりフラッシュ済みにセットされていれば、前記メモリエレメントへの書き戻しを行なわないことを特徴とする請求項 1 記載の電子計算機。

2

## 【請求項 3】 前記書き戻し手段は、

前記フラッシュ手段によるキャッシュフラッシュの実行中に前記処理手段からアクセス要求があり、前記キャッシュメモリに対してキャッシュアロケートを行なう場合に、前記ステータスに基づいて有効でないブロックまたはダーティでないブロックを優先的に追い出し、

追い出すべきブロックが存在しない場合に、前記処理手段によるアクセス対象とするブロックがフラッシュの完了したブロック以外であって、フラッシュ完了ビット F がフラッシュ済みにセットされていないキャッシュブロックをメモリエレメントに書き戻すことを特徴とする請求項 2 記載の電子計算機。

【請求項 4】 前記メモリエレメントが 2 重化された電子計算機であって、

前記フラッシュ手段は、

第 1 のメモリエレメントに対してキャッシュフラッシュを行なう時は、前記処理手段からの全てのライトアクセス要求に対して、フラッシュ完了ビット F をフラッシュ済みにセットし、

第 2 のメモリエレメントに対するキャッシュフラッシュを前記フラッシュ完了ビット F を含む前記ステータスに基づいてキャッシュフラッシュを行なうことを特徴とする請求項 2 または請求項 3 記載の電子計算機。

【請求項 5】 演算処理を行なう処理手段と、前記処理手段によって用いられる複数のキャッシュブロックに分割されたキャッシュメモリと、前記キャッシュメモリを制御するキャッシュ制御手段とを備えたプロセッサエレメントが、前記処理手段により扱われる命令及びデータを格納するメモリエレメントとシステムバスを介して接続された電子計算機において、

前記キャッシュ制御手段は、

前記キャッシュメモリを管理するステータスとして、キャッシュブロックに有効なデータがあることを示す有効ビット V、キャッシュブロックに格納されているデータが更新されていることを表すダーティビット D を用いて管理を行なうものであって、

前記処理手段からの特殊命令に応じて、全キャッシュブロックに関する前記キャッシュステータスを検査して、データが更新されたキャッシュブロックのデータを前記メモリエレメントに書き戻すキャッシュフラッシュを実行すると共に、キャッシュフラッシュ中であることを表すフラッシュ実行通知を行ない、

キャッシュフラッシュの実行中に前記処理手段によりアクセス要求があった場合に、キャッシュフラッシュを中断してフラッシュ中断通知を行ない、

前記処理手段によるキャッシュフラッシュ動作中のアクセス処理を実行可能にすることを特徴とするキャッシュ制御方法。

【請求項 6】 前記キャッシュ制御手段は、

新たにキャッシュブロックの更新されたデータをメモリ

エレメントに書き戻したことを表すフラッシュ完了ビットFを、さらに前記ステータスとして設けて前記キャッシュメモリを管理するもので、

キャッシュフラッシュ実行中に前記処理手段によるライトアクセス要求があった場合に、

前記キャッシュメモリの当該キャッシュブロックが、既にダーティブロックであり、フラッシュの中断前にキャッシュフラッシュが完了したブロックである時、

当該キャッシュブロックのデータを前記メモリエレメントに書き戻し、

当該キャッシュブロックに対応するステータスのフラッシュ完了ビットをフラッシュ済みにセットし、

フラッシュ完了ビットFがフラッシュ済みにセットされているキャッシュブロックに対しては前記メモリエレメントへの書き戻しを行なわないことを特徴とする請求項5記載のキャッシュ制御方法。

【請求項7】 キャッシュフラッシュの実行中に前記処理手段からアクセス要求があり、前記キャッシュメモリに対してキャッシュアロケートを行なう場合に、前記ステータスに基づいて有効でないブロックまたはダーティでないブロックを優先的に追い出し、

追い出すべきブロックが存在しない場合に、前記処理手段によるアクセス対象とするブロックがフラッシュの完了したブロック以外であって、フラッシュ完了ビットFがフラッシュ済みにセットされていないキャッシュブロックをメモリエレメントに書き戻すことを特徴とする請求項6記載のキャッシュ制御方法。

【請求項8】 前記メモリエレメントが2重化された電子計算機であって、

第1のメモリエレメントに対してキャッシュフラッシュを行なう時は、前記処理手段からの全てのライトアクセス要求に対して、フラッシュ完了ビットFをフラッシュ済みにセットし、

第2のメモリエレメントに対するキャッシュフラッシュを前記フラッシュ完了ビットFを含む前記ステータスに基づいてキャッシュフラッシュを行なうことを特徴とする請求項6または請求項7記載のキャッシュ制御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、キャッシュメモリを有する計算機に関する。

【0002】

【従来の技術】 多重化計算機においては、プログラムの実行の途中経過（チェックポイント）をメインメモリに保存しておき、障害が発生したときにチェックポイントから再実行をはかるチェックポイントリスタート方式を採用して、システムダウンを回避する構成としたものがある。

【0003】 チェックポイントリスタート方式については、例えばUSP4, 819, 154号、USP4, 6 50

54, 819号、特開昭59-160899号、また論文Philip A. Bernstein, "Sequoia: A Fault-Tolerant Tightly Coupled Multiprocessor for Transaction Processing," IEEE Computer, Feb. 1988, pp.37-45. に記載されている。

【0004】 図6には、チェックポイントリスタート方式を採用する多重化計算機のシステム構成の一例を示している。図6に示す多重化計算機は、2つのプロセッサエレメント（PE1）10a, （PE2）10bと、1つのメモリエレメント（ME1）12が、システムバス14を介して相互に接続されて構成されている。それぞれのプロセッサエレメント（PE1）10a, （PE2）10bは、独立に動作している。各プロセッサエレメント（PE1）10a, （PE2）10bには、MPU16a, 16b、キャッシュメモリ18a, 18b、及びキャッシュコントローラ19a, 19bが設けられている。キャッシュメモリ18a, 18b内のデータの管理は、それぞれ専用のハードウェアであるキャッシュコントローラ19a, 19bによって行なう。また、MPU16a, 16bには、複数のレジスタが存在し、図中R1, ..., Rnによって示している。

【0005】 また、図7には、チェックポイント方式において障害が発生した場合の処理の流れを示している。図7に示す流れに従って、図6に示す多重化計算機の動作について以下に説明する。

【0006】 チェックポイント処理を行なう時間t(n)になると、プロセッサエレメント（PE1）10aのMPU18aは、一時処理を中断し、全てのレジスタの内容R1, ..., Rnをメモリエレメント（ME1）12の固定領域C1に直接書き込む。そして、キャッシュの更新されたデータDB1, ..., DBkを、メモリエレメント（ME1）12に書き戻す（図7, P0）。

【0007】 チェックポイント処理が完了すると、中断していた処理を再開する。その後、障害Xがプロセッサエレメント（PE1）10aで発生すると、プロセッサエレメント（PE1）10aをシステム構成から切り離し、プロセッサエレメント（PE2）10bによって、プロセッサエレメント（PE1）10aが行っていた処理を引き継ぐ。

【0008】 この時、プロセッサエレメント（PE2）12bは、チェックポイントt(n)で、メモリエレメント（ME1）12に保存されたプログラムの状態から処理を再開する。

【0009】 プロセッサエレメント（PE2）10bは、プロセッサエレメント（PE1）10aが行っていた処理をP0の時点から再開するので、図7中に示すP0からXまでにプロセッサエレメントPE1が行っていた処理結果が無効化されなければならない。

【0010】 これは、チェックポイント以外では、更新されたキャッシュブロックCBのデータをメモリエレ

5

ント (ME1) 12に書き戻さないようにして実現する。このため、チェックポイント処理を頻繁に行なわないように、キャッシュ方式は、例えばnウェイ・セット・アソシエティブ方式をとり、数MBの大容量のキャッシュにするのが一般的である。

【0011】この種、キャッシュ方式においては、キャッシュブロックのデータをメモリエlementに書き戻す必要がある場合には、MPUに対して割り込みが発生される。MPUは、データをメモリエlementに書き戻す処理、すなわちフラッシュ動作を実行させるためのフラッシュ特殊命令FCを、キャッシュコントローラに出力する。

【0012】キャッシュコントローラは、フラッシュ動作を開始し、キャッシュメモリに格納されたデータをキャッシュブロック単位で管理するためのキャッシュタグ(データが有効であることを示す有効ビットV、及びデータが書き替えられたことを示すダーティビットDを含む)を順次検索する。ここで、書き替えられたデータ、すなわちダーティビットDが「1」(書き替えられていることを示す)のブロックがあればメモリエlementに書き戻す。この動作を、キャッシュの全領域について実施する。

【0013】従来では、前述のようなフラッシュ動作が実行されている間に、MPUからのアクセス要求があった場合には、MPUは、フラッシュ動作が完了するまで待っていないなければならない。すなわち、フラッシュ動作中に、MPUからのアクセス要求に応じて、キャッシュメモリ中のデータを変更してしまうと、メモリエlementに書き戻したデータとの一貫性が失われてしまうことがあるためである。

【0014】

【発明が解決しようとする課題】このように従来の多重化計算機においては、フラッシュ動作中にはデータのー貫性を保つために、MPUからの処理要求があったとしても、MPUを待機させる必要があった。フラッシュ機構はハードウェアにより実現されているが、キャッシュの全ブロックのタグを検索しなければならないので非常に時間がかかってしまう。従って、フラッシュ動作中に処理を要求したMPUは、非常に長い時間待機させられてしまうことがあった。

【0015】本発明は前記のような事情を考慮してなされたもので、キャッシュブロックのデータをメインメモリに書き戻すフラッシュ動作中に発生したMPUからの処理要求を実行することが可能な計算機を提供することを目的とする。

【0016】

【課題を解決するための手段及び作用】本発明は、演算処理を行なう処理手段と、前記処理手段によって用いられる複数のキャッシュブロックに分割されたキャッシュメモリと、前記キャッシュメモリを制御するキャッシュ

6

制御手段とを備えたプロセッサelementが、前記処理手段により扱われる命令及びデータを格納するメモリエlementとシステムバスを介して接続された電子計算機において、前記キャッシュ制御手段は、前記キャッシュメモリを管理するステータスとして、キャッシュブロックに有効なデータがあることを示す有効ビットV、キャッシュブロックに格納されているデータが更新されていることを表すダーティビットDを用いて管理を行なうものであって、前記処理手段からの特殊命令に応じて、全キャッシュブロックに関する前記キャッシュステータスを検査して、データが更新されたキャッシュブロックのデータを前記メモリエlementに書き戻すキャッシュフラッシュを実行すると共に、キャッシュフラッシュ中であることを表すフラッシュ実行通知、及び前記処理手段からのアクセス要求によってキャッシュフラッシュを中断していることを表すフラッシュ中断通知を行なうフラッシュ手段を備え、前記フラッシュ手段は、前記フラッシュ実行通知がフラッシュ実行中を示す際に前記処理手段によりアクセス要求があった場合に、キャッシュフラッシュを中断してフラッシュ中断通知を行ない、前記処理手段によるアクセス処理を実行可能にすることを特徴とする。

【0017】これにより、フラッシュ実行中であっても、処理手段(MPU)からのアクセス要求を優先して実行させることができるので、処理手段を待ち状態にすることが回避される。

【0018】また、前記キャッシュ制御手段は、新たにキャッシュブロックの更新されたデータをメモリエlementに書き戻したことを表すフラッシュ完了ビットFを、さらに前記ステータスとして設けて前記キャッシュメモリを管理するもので、前記処理手段からのライトアクセス要求があった場合に、前記ライトアクセス要求の対象となるキャッシュブロックに対してライト処理を実行するライト処理手段と、前記ライト処理手段によるライト処理の対象となるキャッシュブロックが、ダーティブロックであり、前記フラッシュ手段によるフラッシュの中断前にキャッシュフラッシュが完了したブロックではない場合に、当該キャッシュブロックのデータを前記メモリエlementに書き戻し、当該キャッシュブロックに対応するステータスのフラッシュ完了ビットをフラッシュ済みにセットする書き戻し手段と、をさらに具備し、前記処理手段は、前記書き戻し手段によりフラッシュ済みにセットされていれば、前記メモリエlementへの書き戻しを行なわないことを特徴とする。

【0019】これにより、キャッシュフラッシュが中断された際のアクセス要求の対象ブロックが、フラッシュ中断前にフラッシュが完了しているか否かで、データの不整合が生じないように管理される。

【0020】また、前記書き戻し手段は、前記フラッシュ手段によるキャッシュフラッシュの実行中に前記処理

手段からアクセス要求があり、前記キャッシュメモリに対してキャッシュアロケートを行なう場合に、前記ステータスに基づいて有効でないブロックまたはダーティでないブロックを優先的に追い出し、追い出すべきブロックが存在しない場合に、前記処理手段によるアクセス対象とするブロックがフラッシュの完了したブロック以外であって、フラッシュ完了ビットFがフラッシュ済みにセットされていないキャッシュブロックをメモリエlementに書き戻すことを特徴とする。

【0021】これにより、キャッシュオーバフローが発生する場合には、フラッシュ完了ビットFを参照して本来書き戻す必要があるブロックを追い出すことにより、フラッシュ中断に関係なくデータ整合性が保たれる。

【0022】さらに前記メモリエlementが2重化された電子計算機であって、前記フラッシュ手段は、第1のメモリエlementに対してキャッシュフラッシュを行なう時は、前記処理手段からの全てのライトアクセス要求に対して、フラッシュ完了ビットFをフラッシュ済みにセットし、第2のメモリエlementに対するキャッシュフラッシュを前記フラッシュ完了ビットFを含む前記ステータスに基づいてキャッシュフラッシュを行なうことを特徴とする。

【0023】これにより、メモリエlementが2重化されたシステムであっても、各メモリエlementに対するキャッシュフラッシュが中断されても、データの整合性を保ちながら処理手段(MPU)のアクセス要求を実行させることができる。

【0024】

【実施例】以下、図面を参照して本発明の一実施例を説明する。図1は本実施例に係わる多重化計算機におけるキャッシュコントローラの詳細な構成を説明するためのブロック図である。多重化計算機の概略構成は、従来技術の説明で用いた図6と同じであるものとする。図1では1つのプロセッサelement(PE1)10aとメモリエlement(ME1)12のみを示している。

【0025】プロセッサelement10aには、MPU16a、キャッシュメモリ(CM)18a、及びキャッシュコントローラ51が設けられている。なお、他のプロセッサelement(PE2)10bにもキャッシュコントローラ51と同様の構成によるキャッシュコントローラが設けられているものとする。

【0026】本実施例において、キャッシュメモリ(CM)18aは、32Kライン、4ブロック/ライン、32バイト/ブロック構成であるものとする。MPU16aが出力するMPUアドレスMAは、図2に示すように、32ビットであるものとする。キャッシュメモリ18a中のキャッシュブロックCBは、MPUアドレスMAのビット18から5(ブロックアドレスBA)でアドレスされ、キャッシュブロックCB内の各バイトは、MPUアドレスMAのビット4から0でアドレスされる。

MPUアドレスMAのビット31から19(アドレスタグTA)は、後述するキャッシュタグによってキャッシュのヒット/ミス判定のために用いられる。

【0027】キャッシュコントローラ51は、キャッシュメモリ18aのデータの管理を行なう専用のハードウェアである(詳細については後述する)。なお、本実施例では、4ウェイ・セット・アソシアティブ方式を用いているものとする。

【0028】図1に示すように、キャッシュコントローラ51は、タグ(TAG)ブロック52、プロセッサインタフェース(PIF)ブロック53、リードオペレーション(ROP)ブロック54、ライトオペレーション(WOP)ブロック55、キャッシュアロケート(CA)ブロック56、コピーバック(WB)ブロック57、フラッシュオペレーション(FOP)ブロック58、システムバスインタフェース(SIF)ブロック59、及びマルチプレクサ(MUX)60によって構成されている。

【0029】TAGブロック52は、キャッシュメモリ18a内のデータを管理するためのキャッシュタグを記憶しておくものである。キャッシュタグの構成を、図3に示す。キャッシュタグは、MPUアドレスMAのブロックアドレスBAでアドレスされ、一度に1ライン分のブロック(4ブロック)が読み出される。1ライン中の各ブロックのキャッシュタグは、TAG0、TAG1、TAG2、TAG3で区別される。また、キャッシュタグは、アドレスタグTAとキャッシュステータスSTATで構成される。アドレスタグTAには、MPUアドレスMAのビット31から19が格納されており、アクセスの対象とするデータがキャッシュメモリ18aに存在するか否か(キャッシュのヒット/ミス)の判定に使用される。キャッシュステータスSTATは、ブロックが有効であることを表す有効ビットV(V=1で有効を示す)、ブロックが更新されていることを表すダーティビットD(D=1でダーティを示す)、先行してキャッシュメモリ18a中のデータのフラッシュが完了したことを表すフラッシュビットF(F=1でフラッシュ完了を示す)からなる。以下の説明では、単にTAG、TA、STATと表した場合、キャッシュにヒットしたブロックを示すものとする。TAGi、TAi、STATiと表した場合は、1ライン全てのタグを示すものとする(i=0, 1, 2, 3)。

【0030】PIFブロック53は、MPU16aとのインタフェースを司るもので、MPUアドレスMA、各種命令(リードアクセス要求RD、ライトアクセス要求WD、フラッシュ特殊命令FC等)の入力、及びデータの授受を行なう。PIFブロック53は、MPU16aからリードアクセス要求(RD)があるとROPブロック54を起動し、MPU16aからライトアクセス要求(WR)とWOPブロック55を起動する。また、フラ

ッシュ特殊命令FCがMPU16aから発行されると、PIFブロック53は、FOPブロック58を起動して、フラッシュ処理を実行させる。また、MPU16aからのアクセス要求があった場合に、この旨を示すMPUアクセス信号DP (DP=1でアクセス要求有りを示す) をFOPブロック58に通知する。すなわち、MPUアクセス信号DPにより、FOPブロック58によるフラッシュ動作を中断させて、アクセス要求の処理を実行できるようにする。

【0031】ROPブロック54は、PIFブロック53によって起動され、キャッシュヒット (TAGブロック52からのHit=1の通知) 時に、TAGブロック52から得られるタグ (TA, STAT) をもとに、キャッシュメモリ18aからデータの読み込みを行なう。キャッシュミス時には、CAブロック56を起動して、メモリエレメント12からデータの読み込みを実行させる。

【0032】WOPブロック55は、PIFブロック53によって起動され、キャッシュヒット (TAGブロック52からのHit=1の通知) 時にTAGブロック52から得られるタグ (TA, STAT) をもとに、キャッシュメモリ18aに対する書き込みを行なう。キャッシュミス時には、CAブロック56を起動して、メモリエレメント12からデータの読み込みを実行させる。

【0033】CAブロック56は、ROPブロック54、及びWOPブロック55から起動され、キャッシュミス時に、有効なデータをSIFブロック59を介してメモリエレメント12から読み出してキャッシュメモリ18aに書き込む。

【0034】WBブロック57は、FOPブロック58、WOPブロック55、またはCAブロック57から起動され、キャッシュメモリ18aの書き戻すべきデータ (キャッシュステータスSTAT (D) が「1」) をSIFブロック59を介してメモリエレメント12に書き戻す (コピーバック)。

【0035】FOPブロック58は、フラッシュ処理を制御するもので、TAGブロック52に格納されたタグに基づいて、WBブロック57を用いてキャッシュメモリ18a中の書き戻すべきデータをメモリエレメント12にフラッシュするものである。FOPブロック58は、MPU16aからフラッシュ特殊命令FCが出力された際に、PIFブロック53によって起動され (FC通知)、MPUアドレスMAに応じてTAGブロック52からキャッシュタグTAGi (i=0, 1, 2, 3) を読出し、タグアドレスTA、キャッシュステータスSTAT (有効ビットV、ダーティビットD、フラッシュビットF) の検査を行なう。また、FOPブロック58は、FC通知に応じてフラッシュ操作を開始した際にはフラッシュ実行信号DF (DF=1でフラッシュ実行中を示す) を、またフラッシュ操作中にMPU16aか

らのアクセス要求 (MPUアクセス信号DP=1) に応じてフラッシュ操作を中断した際にはフラッシュ中断信号AF (AF=1でフラッシュ中断を示す) を、PIFブロック53を介してROP54またはWOPブロック55に出力する。なお、FOPブロック58には、TAGブロック52の検索すべき位置 (ライン) を示す検索アドレスカウンタ (SA) 58aが設けられている。

【0036】システムバスインタフェース (SIF) ブロック59は、メモリエレメント12とのインタフェースを司るもので、ROPブロック54、WOPブロック55、CAブロック56、WBブロック57と接続されている。

【0037】マルチプレクサ (MUX) 60は、MPU16aからのMPUアドレスMA (ブロックアドレスBA)、及びFOPブロック58からのMPUアドレスMAを入力し、択一的にTAGブロック52に供給するものである。通常は、MPU16aからのMPUアドレスMAを選択しており、必要に応じてFOPブロック58からのMPUアドレスMAを選択して供給する。

【0038】次に、本実施例における計算機システムの動作について説明する。はじめに、フラッシュ動作について説明する。まず、MPU16aからのフラッシュ特殊命令FCを、キャッシュコントローラ51は、PIFブロック53によって受けとる。PIFブロック53は、FOPブロック58を起動し (FC通知)、フラッシュ動作を開始させる。一方で、MPU16aに対しては、バストランザクションを完了させる。

【0039】FOPブロック58は、起動されると検索アドレスカウンタ (SA) 58aを「0」にクリアし、フラッシュ実行信号DFを「1」にセットする。フラッシュ実行信号DFは、フラッシュ操作が完了するまで「1」に保持される。

【0040】次に、FOPブロック58は、TAGブロック52から、検索アドレスカウンタ (SA) 58aが指すキャッシュタグTAGi (i=0, 1, 2, 3) を読出し、キャッシュステータスの有効ビットSTATi (V) (i=0, 1, 2, 3)、及びダーティビットSTATi (D) (i=0, 1, 2, 3) とフラッシュビットSTATi (F) (i=0, 1, 2, 3) を検査する。

【0041】ここで、各ブロックのフラッシュビットFが「0」、有効ビットVが「1」で、かつダーティビットDが「1」であれば、そのキャッシュブロックCBが更新されているため (ダーティブロック)、FOP58は、WBブロック57を起動する。WBブロック57は、該当するTAGi (i=0, 1, 2, 3) に応じて、該当するブロックのデータをキャッシュメモリ18aから読出し、SIFブロック59を介してメモリエレメント12に書き戻す (コピーバック)。

【0042】1ライン中の各ブロックについてコピーバ



11

ックが完了すると、FOPブロック58は、検索アドレスカウンタ(SA)58aをカウントアップ(+1)する。また、1ライン中の全てのブロックのダーティビットDが「0」の時は、FOPブロック58は、メモリエlement12に書き戻すべきデータがないので、検索アドレス(SA)58aをカウントアップ(+1)するだけで、次のラインに検索対象を移す。

【0043】この時、キャッシュステータスSTATのフラッシュ完了ビットSTATi(F)が「1」であれば、すでにそのブロックはメモリエlementにコピーバックされているので、ダーティビットD=1であってもコピーバックを行わずに、フラッシュ完了ビットSTATi(F)を「0」にクリアする。

【0044】以上の処理をキャッシュの全領域、すなわちTAGブロック52の全てのキャッシュタグを検索し、各キャッシュブロックについて実施する。ところで、前述のようなフラッシュ動作の実行中に、MPU16aからのアクセス要求があった場合、あるトランザクションが完了した切りが良い時点、例えばFOPブロック58が検索アドレス(SA)58aをカウントアップ(+1)した後で、フラッシュ中断信号AFを「1」にセットしフラッシュ動作を中断する。

【0045】その後、MPU16aからのアクセス要求に対する処理が終了すると(DP=0)、フラッシュ中断信号AFを「0」にクリアしフラッシュ動作を再開する。すなわち、本発明のシステムでは、フラッシュ動作中であってもMPU16aからのアクセス要求(リードアクセス要求RD、ライトアクセス要求WD)がある場合には、MPU16aをフラッシュ動作が完了するまで待機させるのではなく、アクセス要求を優先させて実行できるようにフラッシュ動作を中断させる。

【0046】MPU16aからアクセス要求があると、PIFブロック53は、MPUアクセス信号DPを「1」にセットする。この時、フラッシュ実行中(DP=1)であれば、フラッシュ動作が中断されるのを待って(AF=1)、要求された処理を起動させる。

【0047】フラッシュが中断されると(AF=1)、MPU16aアクセスを行なう。なお、リードアクセス要求RD、及びライトアクセス要求WDがあった場合のそれぞれの動作については後述する。

【0048】MPU16aアクセスが完了すると、MPU16aアクセス信号DPを「0」にクリアし、フラッシュ動作が再開される。次に、MPU16aからリードアクセス要求RDがあった場合について説明する。

【0049】MPU16aからリードアクセス要求RDがあると、PIFブロック53は、MPUアクセス信号DPを「1」にセットし、フラッシュ動作の中断をFOPブロック58に要求する。

【0050】MPUアクセス信号DPによりフラッシュ動作の中断を要求すると、フラッシュ実行中であれば、

12

FOPブロック58によって、前述したようなフラッシュ中断のための処理が実行されフラッシュ中断信号AFが「1」にセットされる。

【0051】フラッシュ動作が中断または完了すると(AF=1 or DP=0)、FOP58は、TAGブロック52から、検索アドレスカウンタ(SA)58aが指すキャッシュタグTAGiを読み出す。

【0052】ここで、キャッシュにヒットした場合(Hit=1)、ROP54は、ヒットしたTAG(TA, STAT)に応じて、キャッシュメモリ18aからデータを読み出してMPU16aへデータを送信する。そして、ROP54は、フラッシュ動作が再開できるようにMPUアクセス信号DPを「0」にセット(PIFブロック53を介してFOPブロック58に通知)する。また、FOP58は、フラッシュ中断信号AFを「0」にセットする。

【0053】一方、キャッシュにミスヒットした場合(Hit=0)、ROP54は、メモリエlement12からデータを読み込み、キャッシュアロケートするためにCAブロック56を起動し、SIFブロック59を介してメモリエlement12からデータを読み出す。

【0054】CAブロック56は、タグブロック53のブロックアドレスBAにより指定されるライン中の全てのブロックの有効ビットSTATi(V)(i=0, 1, 2, 3)と、ダーティビットSTATi(D)(i=0, 1, 2, 3)を検査する。

【0055】1ライン内に有効ビットSTATi(V)が「1」にセットされていないブロックがあれば、そのブロックのデータは不要なので、同ブロックに対しキャッシュアロケートを行なう。

【0056】1ライン内の全てのブロックの有効ビットSTATi(V)が「1」にセットされている場合、何れかのブロックをキャッシュから追い出す必要がある。通常は、MPU16aにキャッシュオーバフローを通知してフラッシュ操作を行なうが、フラッシュ動作中のアクセス要求に対する処理の場合であれば、次のようにして何れかのブロックをキャッシュから追い出す。

【0057】一般に何れのブロックを追い出すかは、例えばランダム法やLRU(Least Recently Used)法などに基づいて決定するが、本発明におけるキャッシュ追い出しの方法は、さらにダーティビットSTATi(D)が「1」にセットされていない(メモリエlement12にデータを書き戻す必要がない)ブロックを優先的に追い出してキャッシュアロケートを行なう。

【0058】該当するブロックが存在せず(全てのダーティビットSTATi(D)が「1」)、検索アドレス(SA)58aがブロックアドレスBAより小さい(SA<BA)場合、すなわち図4(a)に示すように、対象とするブロックのフラッシュが完了されておらず、フラッシュビットSTATi(F)が「0」のブロックが

あれば、WBブロック57は、メモリエlement 12にそのブロックのデータを書き戻す(コピーバック)。そして、そのキャッシュブロックにキャッシュアロケートを行なう。

【0059】これに対して、図4(b)に示すように、検索アドレスSAがブロックアドレスBA以上で( $SA \geq BA$ )、1ライン内の全ブロックのダーティビットSTATi(D)が「1」の時、または検索アドレスSAがブロックアドレスBAより小さく( $SA < BA$ 、フラッシュが完了していない)、1ライン内の全てのブロックのフラッシュビットSTATi(F)が「1」でかつ、ダーティビットSTATi(D)が「1」の場合には、キャッシュアロケートは失敗となる。

【0060】この場合、MPU16aへキャッシュオーバーフローが通知され、現在のフラッシュ動作が完了するまで、MPU16aの全ての処理は待たされる。現在のフラッシュ動作が完了すると、再度フラッシュ命令を実行する必要がある。

【0061】キャッシュアロケートが成功すると、ROPブロック54は、MPU16aが要求したデータをキャッシュメモリ18aから読み出して送信する。次に、MPU16aからライトアクセス要求WDがあった場合について説明する。

【0062】MPU16aからライトアクセス要求WRがあると、PIFブロック53は、MPUアクセス信号DPを「1」にセットし、フラッシュ動作の中断をFOPブロック58に要求する。

【0063】MPUアクセス信号DPによりフラッシュ動作の中断を要求すると、フラッシュ実行中であれば、FOPブロック58によって、前述したようなフラッシュ中断のための処理が実行され、フラッシュ中断信号AFが「1」にセットされる。

【0064】フラッシュ動作が中断または完了すると( $AF=1$  or  $DF=0$ )、FOP58は、TAGブロック52から、検索アドレスカウンタ(SA)58aが指すキャッシュタグTAGiを読み出す。

【0065】ここで、キャッシュにヒットした場合( $Hit=1$ )でかつ、そのキャッシュステータスのフラッシュビットSTAT(F)が「1」なら、WOP55は、キャッシュメモリ18aに対してライトを行なう。キャッシュヒットでフラッシュが完了したブロック( $SA \geq BA$ )に対するライト動作は成功する。

【0066】キャッシュにヒットした場合( $Hit=1$ )でかつ、そのブロックのフラッシュビットSTAT(F)が「0」でダーティビットSTAT(D)が「1」の時、検索アドレスSAがブロックアドレスBAより小さい( $SA < BA$ )なら、そのブロックはメインメモリへフラッシュされていないので、メモリエlement(ME1)12へそのブロックのデータを書き戻し(コピーバック)、フラッシュビットSTAT(F)を

「1」にセットする。

【0067】一方、キャッシュにミスヒットした場合( $Hit=0$ )は、WOP55は、キャッシュアロケートを行なうため、CAブロック56を起動し、SIFブロック59を介してメモリエlement 12からデータを読み出す。

【0068】CAブロック56は、前述したリードアクセス要求RDに対してキャッシュにミスヒットした( $Hit=0$ )場合と同様にして、キャッシュアロケートを行なう。

【0069】キャッシュアロケートが成功すると、CAブロック56は、MPU16aからのライトを行ない、ダーティビットDと、検索アドレスSAがブロックアドレスBAより小さい( $SA < BA$ )なら、キャッシュステータスSTATのフラッシュビットFも「1」にセットする。

【0070】MPU16aからのライトアクセスが完了すると、PIFブロック53は、MPUアクセス信号DPを「0」にセットし、フラッシュ動作を再開させる。同時に、FOP58は、再度TAGブロック52からキャッシュタグを読み出し、キャッシュステータスSTATを検査する。

【0071】検索アドレスSAがブロックアドレスBAより小さく( $SA < BA$ )、1ライン全てのキャッシュブロックの有効ビットSTATi(V)とダーティビットSTATi(D)、及びフラッシュ完了ビットSTATi(F)が全て「1」の時、または検索アドレスSAがブロックアドレスBA以上( $SA \geq BA$ )で、1ライン全てのキャッシュブロックの有効ビットSTATi(V)とダーティビットSTATi(D)が全て「1」の時は、MPU16aに対してキャッシュオーバーフロー割り込みを発生させる。

【0072】このようにして、キャッシュコントローラ51は、MPU16aからのフラッシュ特殊命令FCを受けると、FOPブロック58によってキャッシュタグのキャッシュステータスSTATを検査して、データが更新されたブロックについてメモリエlementに書き戻す。その際、MPU16aからリードアクセス要求RDあるいはライトアクセス要求WDがあると、キャッシュフラッシュ動作を中断して、フラッシュ中断信号AFによって通知し、アクセス要求を実行可能な状態にする。これにより、フラッシュ動作を中断させて、MPU16aのアクセス要求を優先させて実行させることができる。

【0073】また、MPU16aは、キャッシュコントローラ51によるフラッシュ動作が中断されると、リードアクセスまたはライトアクセスを実行する。ライトアクセスを行なう場合、キャッシュの対象ブロックが既にダーティ( $D=1$ )で、フラッシュ動作の中断前にフラッシュが完了していなければ( $SA < BA$ )、メモリエ

レメント18aに書き戻して、キャッシュステータスのフラッシュ完了ビットFを「1」にセットしておく。そして、フラッシュ動作を再開した際には、フラッシュ完了ビットFが「1」のブロックについてはメモリエlement18aへの書き戻しを行なわないことにより、フラッシュ中断に関係なくデータ整合性を保つことができる。

【0074】また、フラッシュ動作中のMPU16aからのアクセス要求によって、キャッシュアロケートが実行される際に、キャッシュオーバフローが発生する場合には、追い出すブロックを基本的にはランダム法やLRU法などに基づいて決定するが、さらに有効でないブロック(V=0)、ダーティでないブロック(D=0)を優先的に追い出し、これらのブロックが存在しなければ(V=1, D=1)、フラッシュ動作中断前のフラッシュ動作でフラッシュが完了しておらず、フラッシュ完了ビットFが「1」にセットされていなければ、すなわち本来書き戻す必要があるブロックをメモリエlement12に書き戻してキャッシュアロケートを行なう。これにより、フラッシュ中断に関係なくデータ整合性を保つことができる。

【0075】なお、前述した実施例は、図6に示すように、1個のメモリエlement(ME1)が設けられた設けられた構成として説明しているが、図5に示すようにメモリが2重化されている構成の場合には、2度に渡ってキャッシュフラッシュ動作を行なう必要(それぞれのメモリに対するフラッシュ動作)がある。

【0076】1回目のフラッシュ動作の間にMPUアクセスがあると、ライトヒットで、キャッシュステータスが既にダーティ(STATI(D)=1)で、フラッシュビットSTAT(F)が1の時だけ、MPU16aは待たされる。それ以外では、MPU16aは、キャッシュコントローラのフラッシュ動作を中断させて、リードアクセス要求、ライトアクセス要求を実行することができる。

【0077】この場合、検索アドレスSAと、MPU16aからのアクセスのブロックアドレスBAの大小関係に関わらず(キャッシュ動作中断前に書き戻しが完了しているか否かに関係なく)、ライト動作には、キャッシュフラッシュビットSTAT(F)を「1」にセットする。

【0078】また、1回目のフラッシュ動作中は、フラッシュビットSTATi(F)はクリアしない。従って、2回目のフラッシュ動作時には、フラッシュビットSTATi(F)が「1」にセットされたブロックについての書き戻しは実行されない。2回目のフラッシュ動作は、前述した場合と同様の動作をする。

【0079】このようにして、メモリエlementが2重化されたシステムの場合には、フラッシュ動作を、それぞれのメモリエlementに対して行なう必要があるが、

1つめのメモリエlementに対するフラッシュ動作中にMPU16aからのアクセス要求があってフラッシュ動作が中断され、ライトアクセスされると、全ての対象ブロックのフラッシュ完了ビットFを「1」にセットする。これにより、1つめのメモリエlementに対するフラッシュ動作の中断を伴うライトアクセスの結果を、2つめのメモリエlementに反映させないようにすることができる。1つめのメモリエlementに対しても、フラッシュ動作中断前にフラッシュが完了しているブロック(SA>BA)であれば書き込みは影響せず、フラッシュ動作中断前にフラッシュが完了していないブロック(SA≤BA)であってもフラッシュ完了ビットFを「1」にセットすることで書き戻されない。MPUのライトアクセスによって書き込まれたブロックの内容は、次のフラッシュ動作によってメモリエlementに書き戻される。こうして、メモリエlementが2重化されたシステムであっても、フラッシュ動作を中断させて、MPUのアクセス要求を実行させることができる。

【0080】

【発明の効果】以上詳述したように本発明によれば、フラッシュ動作中にMPUからのアクセス要求があった場合には、フラッシュ動作を中断させて、MPUのアクセスを優先的に実行させるもので、その際にフラッシュ動作中断の前後でデータの整合性が保たれるようにコピーバック、及びキャッシュアロケートを行なうので、MPUは待たされることなく処理を進めることができる。

【図面の簡単な説明】

【図1】本発明の実施例に係わる多重化計算機におけるキャッシュコントローラの詳細な構成を説明するためのブロック図。

【図2】本実施例におけるMPUアドレスMAを説明するための図。

【図3】本実施例におけるキャッシュタグを説明するための図。

【図4】本実施例におけるブロックの追い出しを説明するための図。

【図5】本発明の実施例に係わる多重化計算機の他の構成を示すブロック図。

【図6】多重化計算機の構成の一例を示すブロック図。

【図7】チェックポイント処理を説明するためのシステム構成図。

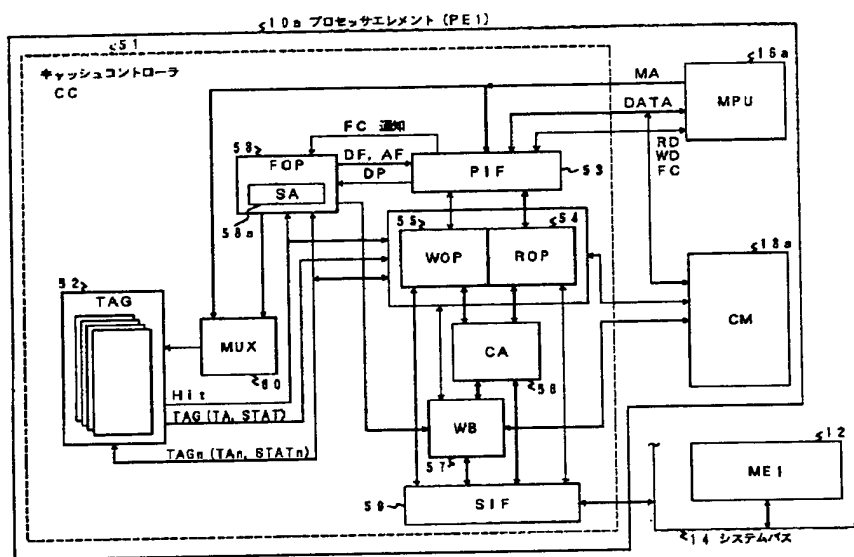
【符号の説明】

10a…プロセッサelement(PE1)、12…メモリエlement(ME1)、16a…MPU、18a…キャッシュメモリ(CM)、52…タグ(TAG)ブロック、53…プロセッサインタフェース(PIF)ブロック、54…リードオペレーション(ROP)ブロック、55…ライトオペレーション(WOP)ブロック、56…キャッシュアロケート(CA)ブロック、57…コピーバック(WB)ブロック、58…フラッシュオペレー

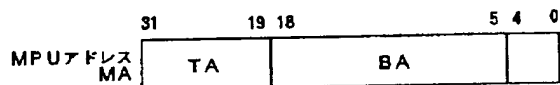
17  
ション (FOP) ブロック、59…システムバスインタ

18  
フェース (SIF) ブロック。

【図1】

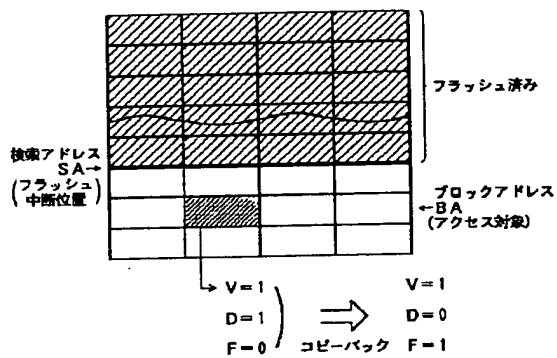


【図2】

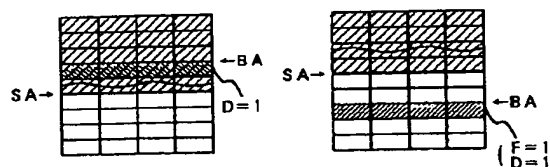


【図4】

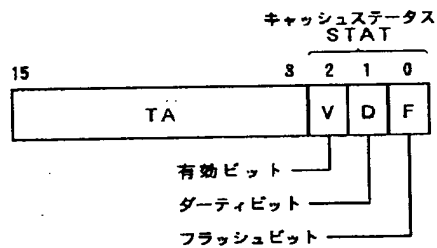
(a)



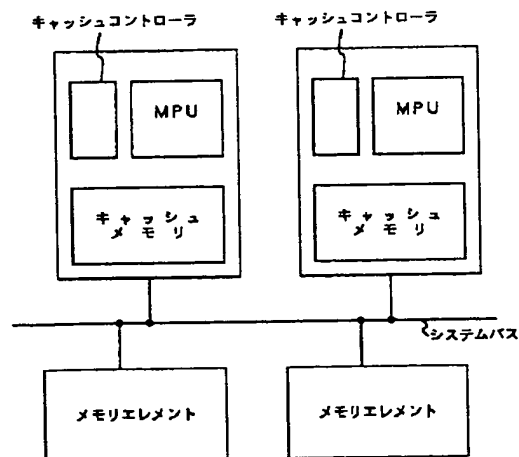
(b)



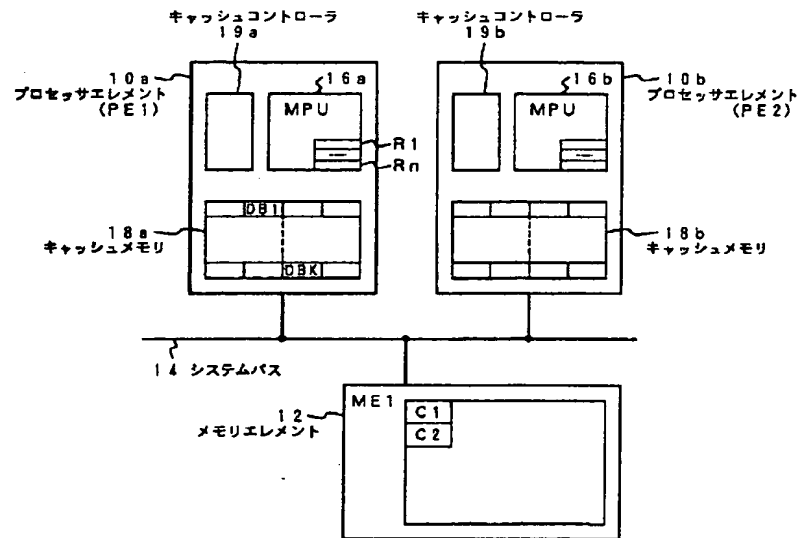
【図3】



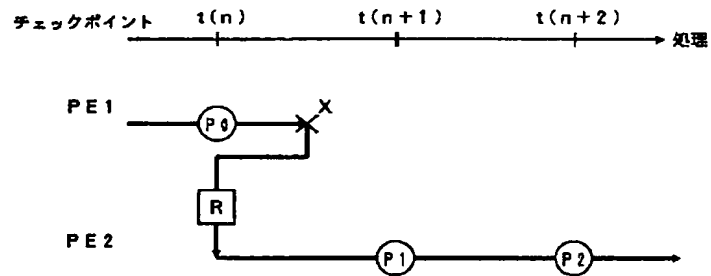
【図5】



【図6】



【図7】



**THIS PAGE BLANK (USPTO)**